

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently amended) A method of processing runtime functions, comprising:
compiling code to produce executable code that is marked with an identifier indicating that the executable code supports runtime protection;
receiving a call to a runtime function of the executable code;
determining associated data from the call to the runtime function;
determining a target address from the associated data;
comparing the target address with a reference list of valid target addresses; ~~and~~
if the target address is found on the reference list of valid target addresses then
executing the ~~target~~ runtime function; and
if the target address is not found on the reference list of valid target addresses then terminating execution of the executable code.
2. (Original) The method of claim 1, wherein the step of determining the associated data comprises accessing data in a data structure connected with the runtime function and calculating the associated data based on the accessed data.
3. (Original) The method of claim 1, further comprising:
determining if at least a portion of the associated data is valid; and
preventing execution of the target if the associated data is not valid.
4. (Canceled)
5. (Currently amended) The method of claim 1 comprising the step of generating the reference list of valid target addresses ~~at compiler and link time~~ during execution of a previous runtime function.
6. (Currently amended) The method of claim 1 comprising the step of generating the reference list of valid target addresses ~~at runtime~~ during execution of a previous runtime

function.

7. (Original) The method of claim 3, wherein the step of determining if the associated data is valid comprises retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies.
8. (Original) The method of claim 3, further comprising determining and storing a predetermined calculated value based on at least a portion of the associated data, prior to receiving the call to the runtime function.
9. (Original) The method of claim 8, wherein determining if the associated data is valid comprises comparing the predetermined calculated value to another calculated value based on the associated data.
10. (Currently amended) A computer-readable medium having stored thereon computer-executable instructions for performing a method of processing runtime functions, the method comprising:
 - receiving a call to a runtime function;
 - determining associated data from the call to the runtime function;
 - determining a target address from the associated data;
 - comparing the target address with a reference list of valid target addresses; and
 - if the target address is found on the reference list of valid target addresses then executing the ~~target~~ runtime function; and
if the target address is not found on the reference list of valid target addresses then terminating execution of the runtime function.
11. (Previously Presented) The computer-readable medium of claim 10, wherein the step of determining the associated data comprises accessing data in a data structure connected with the runtime function and calculating the associated data based on the accessed data.
12. (Original) The computer readable medium of claim 10, having further computer-

executable instructions for determining if at least a portion of the associated data is valid, and preventing execution of the target if the associated data is not valid.

13. (Canceled)

14. (Currently amended) The computer-readable medium of claim 10 comprising the step of generating the list of valid target addresses ~~at compiler and link time~~ during execution of a previous runtime function.

15. (Currently amended) The computer-readable medium of claim 10 comprising the step of generating the list of target addresses ~~at runtime~~ during execution of a previous runtime function.

16. (Original) The computer-readable medium of claim 12, wherein determining if the associated data is valid comprises retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies.

17. (Original) The computer-readable medium of claim 12, having further computer-executable instructions for determining and storing a predetermined calculated value based on at least a portion of the associated data, prior to receiving the call to the runtime function.

18. (Original) The computer-readable medium of claim 17, wherein determining if the associated data is valid comprises comparing the predetermined calculated value to another calculated value based on the associated data.

19. (Currently amended) A system for processing runtime functions, comprising:
a compiler that compiles code to produce an executable that is marked with an identifier indicating that the executable supports runtime protection;
a processor that receives a call to a runtime function; and
a dispatcher system that determines associated data from the call to the runtime function, determines a target address from the associated data, and if the target address is

found on the reference list of valid target addresses then executes the target.

20. (Original) The system of claim 19, wherein the dispatcher system comprises a module to access data in a data structure connected with the runtime function and calculate the associated data based on the accessed data.

21. (Original) The system of claim 19, wherein the dispatcher system comprises modules to determine if at least a portion of the associated data is valid and prevent execution of the target if the associated data is not valid.

22. (Currently amended) The system of claim 21, further comprising a storage device that stores a list of valid targets, wherein the dispatcher system determines if the associated data is valid by comparing the target address to the list of valid ~~targets~~ target addresses.

23. (Previously Presented) The system of claim 19, further comprising a compiler that generates the reference list of valid target addresses.

24. (Original) The system of claim 21, wherein the dispatcher system determines if the associated data is valid by retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies.

25. (Original) The system of claim 21, wherein the processor determines and stores a predetermined calculated value based on at least a portion of the associated data, prior to receiving the call to the runtime function.

26. (Original) The system of claim 25, wherein the dispatcher system determines if the associated data is valid by comparing the predetermined calculated value to another calculated value based on the associated data.

27. (Canceled)

28. (Currently amended) The method of claim 1 comprising the step of storing the target address in a caller provided location during execution of a previous runtime function.

29-30. (Canceled)

31. (Currently amended) The computer-readable medium of claim 10 comprising the step of storing the target address in a caller provided location during execution of a previous runtime function.

32. (Canceled)

33. (Currently amended) The computer-readable medium of claim 10 comprising the step of storing the reference list of valid target addresses in memory during execution of a previous runtime function.

34. (Currently amended) The system of claim 19 comprises a module to store the target address in a caller provided location during execution of a previous runtime function.

35. (Canceled)

36. (Currently amended) The system of claim 19 comprises a module to store the reference list of valid target addresses in memory during execution of a previous runtime function.